Title	A dual cube hashing scheme for solving LPP integrity problem
Author(s)	Fang, J; Jiang, ZL; Yiu, SM; Chow, KP; Hui, LCK; Chen, L; Niu, X
Citation	The 6th International Workshop on Systematic Approaches to Digital Forensic Engineering In conjunction with the IEEE Security and Privacy Symposium (IEEE/SADFE 2011), Oakland, CA., 26 May 2011. In IEEE/SADFE Proceedings, 2011, p. 1-7
Issued Date	2011
URL	http://hdl.handle.net/10722/139989
Rights	IEEE Symposium on Security and Privacy Proceedings. Copyright © IEEE.

A Dual Cube Hashing Scheme for Solving LPP Integrity Problem

Junbin Fang, Zoe L. Jiang, S. M. Yiu, K. P. Chow, Lucas C. K. Hui, Long Chen, Xiamu Niu

Abstract—**In digital forensics, data stored in a hard disk usually contains valuable evidence. Preserving the integrity of the data in the hard disk is a critical issue. A single hash value for the whole hard disk is not appropriate as the investigation may take a long time and latent sector errors (LSEs) (bad sectors due to media imperfection, for example) which cause a sector suddenly unreadable will make the hash value inconsistent. On the other hand, using a hash per sector may need to store a lot of hash values. Previous research has been conducted to use fewer hash values, but can resist some of LSEs to decrease the number of unverifiable sectors even if there are LSEs. This integrity problem is more complicated in the presence of Legal Professional Privileged (LPP) data inside a seized hard disk in digital forensic as the hard disk has to be cloned once seized and the original hard disk will be sealed after cloning. Hash values need to be computed during this cloning process. However, the cloned copy will be returned to the suspect for the deletion of LPP data before the investigator can work on the sanitized copy. Thus, the integrity of unmodified sectors has to be verified using the hash values computed based on the original hard disk. This paper found that existing schemes are not good enough to solve the integrity problem in the presence of both LSEs and deletion of LPP data. We then propose the idea of a "Dual Cube" hashing scheme to solve the problem. The experiments show the proposed scheme performs better than the previous schemes and fits easily into the digital forensic procedure.**

Index Terms—**3-Dimension Hashing, Combinatorial Group Testing (CGT), Hard Disk Sector Allocation, Latent Sector Errors (LSEs), Legal Professional Privilege (LPP)**

I. Introduction

Hard disk, as a most widely-used data storage, plays an important role in digital forensic investigation by providing a huge amount of evidence data. Preserving the integrity of evidence data stored in the hard disk is a fundamental problem. Otherwise, either the *defendant* or the *prosecutor* can easily challenge the validity of it. However, forensic investigation usually involves a lengthy process and it takes time to identify the data inside a huge hard disk that can be used as evidence. Thus, a usual technique is to make use of hash values (a cryptographic technique [1] that can be used to verify if any bit of a piece of digital information has been changed).

Once the hard disk of a suspect is seized, a standard procedure [2-4] is to make a clone using a write-blocking device and create hash values for the data inside the hard disk before any investigation is carried out. The original disk is often sealed and investigation will be carried out on the cloned copy. After the investigation, the integrity of the evidence found in the cloned copy can be verified by computing the hash values of the cloned copy and compare it to the previously stored hash values. However, simply taking one cryptographic hash for the whole hard disk is not appropriate due to the nature of hash function that even if one bit inside the whole hard disk has corrupted, the hash value of the "damaged" hard disk will not be the same as the previously computed hash value, thus the integrity of the hard disk cannot be verified and the evidence becomes useless. In practice, the hard disk sectors may be damaged due to various reasons, such as latent sector errors (LSEs) [5]. Another extreme is to create a hash value for each sector. This approach will create a lot of hash values, for example, for a 250GB hard disk, the number of sectors is over 480 million. Thus, it is desirable to minimize the number of hash values to be stored while increase the chance of maintaining the integrity of a sector. Examples of research in this direction include 3-*D* scheme [6], CGT [7], and super-cube approach [8, 9].

Besides the physical damage to the hard disk that may cause the bits in a sector to be changed or result in an unreadable sector, the integrity problem in digital forensic is more complicated due to the Legal Professional Privilege (LPP). LPP in the Common Law is to enable a client who may not have enough legal knowledge to fully disclose everything to his legal advisor for seeking of advice without worrying that anything disclosed for this purpose will be used against him in the court. Thus, the defendant has a right to prevent a particular document from presenting as evidence against him if it is classified as a privileged document [10]. When it is applied to the digital world, it is possible that LPP data exists in a seized hard disk in

Manuscript received March 14, 2011. The work described in this paper was partially supported by the General Research Fund from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. RGC GRF HKU 713009E), the NSFC/RGC Joint Research Scheme (Project No. N_HKU 722/09), and HKU Seed Fundings for Basic Research 200811159155 and 200911159149.

Zoe L. Jiang is with the School of Computer Science and Technology, Harbin Institute of Technology Shenzhen Graduate School, China (corresponding author : (86)137-1153-3710; e-mail: zoeljiang@gmail.com).

Junbin Fang, S. M. Yiu, K. P. Chow and Lucas C. K. Hui are with the Department of Computer Science, The University of Hong Kong, Hong Kong (e-mails: junbinfang@gmail.com, smyiu@cs.hku.hk, chow@cs.hku.hk, hui@cs.hku.hk).

Long Chen is with the Institute of Computer Science and Technology, Chongqing University of Posts and Telecommunications, China (e-mail: chenlong@cqupt.edu.cn).

Xiamu Niu is with the Faculty of Computer Science, Harbin Institute of Technology, China (e-mail: xiamu.niu@hit.edu.cn).

a criminal investigation. The owner of the hard disk has the right to refuse any possible leakage of the contents of the LPP data.

A hard disk stores numerous files and information (including the logically deleted data). As indicated in [11], in practice, it is not feasible to let the owner delete all LPP data before the suspicious hard disk is being seized and cloned as it requires legal professionals to determine whether a piece of information can be possibly classified as LPP data. The investigator would also worry that the suspect may purposely erase some evidence which is actually non-LPP data. On the other hand, if the investigator is allowed to take a cloned copy without the owner removing LPP data, there is no guarantee that there is no access to the LPP data which may create unfairness to the suspect.

To answer all the above questions, we need to clearly *define a legal and technically-feasible procedure* and *design an effective and flexible integrity verification and identification scheme for a hard disk* when sector change (LSEs or artificial modification/deletion such as removing LPP data) is unavoidable. As a result, when a suspicious hard disk is seized, the procedure can be executed with the permission of the both parties (the hard disk owner and the investigator). Whenever disputation arises, the hard disk integrity verification and identification scheme can be deployed to check the integrity and further point out the sectors which affect the integrity so as to provide proof for any misbehavior of the owner.

From the practical point of view in defining the procedure, we first need to consider the amount of time and effort required, as well as the extra storage and the complexity of the whole procedure, such as the requirement of a face-to-face interaction between the two parties. Secondly, access to the *original* hard disk should be avoided as much as possible, so as to protect the original data from any possible damage. Thirdly but not lastly, the procedure should be fair to both parties. Ref. [11] proposed a procedure trying to satisfy the above requirements. Roughly speaking, the core steps of the procedure are as follow. Once a hard disk is seized, the investigator is allowed to make a cloned copy using a write-blocking device in front of the suspect/owner. Hash values are created during the cloning process. The original hard disk is then sealed and the hash values stored for later verification. The cloned disk will be returned to the owner/suspect. They can spend time to erase the data which can be claimed to be LPP related. Data claimed to be LPP will be submitted to the court. The sanitized hard disk will be give to the investigator for investigation. One of the core problems in the procedure is how to verify the integrity of the data inside the sanitized hard disk after the LPP data has been deleted using the hash values computed from the original hard disk.

Since some sectors will be modified, one single hash value for the whole hard disk will not work. Using one hash for each sector can solve the problem, but it requires storing many hash values. It is desirable to have an intermediate scheme so that the number of hash values to be stored can be reduced while trying to reduce the failure rate to verify the integrity of a sector even if it is not modified. In this paper, we first review and evaluate the effectiveness of existing approaches [6, 7] that were proposed to solve the LPP integrity problem based on simulated LSEs and LPP file deletion. Then, we propose a "Dual Cube" hashing scheme which can greatly reduce the failure rate of verification of the integrity of an unmodified sector while using a lot fewer hash values than the trivial approach of storing one hash value per sector. The rest of the paper is organized as follows. Section II briefly describes some related work. Section III provides the evaluation of existing approaches based on simulated LSEs and deletion of LPP data. From this empirical study, we found that the CGT approach [7] is not appropriate for solving the LPP integrity problem since the performance of it degrades substantially when the number of changed sectors increases no matter whether these changed sectors are consecutive or not. In practice, there should be quite a lot of LPP data will be erased from the hard disk, thus CGT may not be an appropriate solution. On the other hand, based on some observations from the empirical study, we propose a "Dual Cube" hashing scheme based on the 3-*D* scheme [6] in Section IV that can provide a better solution to solve the LPP integrity problem. Section V concludes the paper.

II. Literature Review

Jiang et al. proposed a 3-*D* hashing scheme to provide strong integrity verification function for the hard disks [6]. Instead of computing one chained hash value for a whole hard disk, it first orders all sectors into a 3-dimensional space to form a solid cube (one can imagine the single hash value approach is to arrange all sectors in a 1-dimensional space). Then a hash value is computed on each sector chain in each of the three dimensions. As a result, the integrity of each sector can be verified as long as there is no modified/deleted sector on at least one of its three hash chains, thus it does not depend on all the sectors in the hard disk. However, the design of this scheme assumes that any sector becomes bad after some period of time with only a small probability and does not consider the case of possible artificial modification/deletion, such as LPP data. A failure probability P_f depending on the number of actual bad sectors is given in the paper.

Later, Fang et al. tried to provide a better scheme which uses fewer hash values and try to identify which sector has been changed that causes an unchanged sector not able to be verified. The main idea is to group sectors into different subsets using a sophisticated grouping algorithms based on some Combinatorial Group Testing algorithms [7]. Each group can be treated as a hash chain. A hash value is computed for each group. The scheme relies on a complicated grouping algorithm to form the hash chain, thus the number of hash chains can be significantly reduced, however, the performance of the scheme deteriorates as the number of modified sectors increases (see the details in Section III.C for details).

Chen et al. proposed the *one-error integrity indication code* to compress the storage of hash values for the integrity verification of the data transmitted through network [8]. However, it can only indicate one error sector in grouped

sectors. To enhance error indication capability, they further proposed the *grouped one-error integrity indication code* using super-cube idea [9], which is similar to the 3-*D* hashing scheme [6]. Therefore, it also suffers from the base error amplification ratio, which is similar to the measure of P_f in [6].

Finally, Ref. [11] tried to apply 3-*D* scheme to solve the LPP integrity problem. However, they assume that somehow the list of the sectors (files) to be claimed as LPP data may be known in order to create an additional hash chain on those affected sectors. This assumption is not valid in all cases.

III. Observation on Sector Allocation

In this section, we try to evaluate the existing approaches using simulation based on two types of possible changes in sectors: the incident error sectors and artificial erased files (e.g. due to LPP data). These two types of changes on sectors would affect the performance of designed integrity verification schemes. This section discusses the features of two types of changes, and evaluates the effect of such changes on two existing integrity verification schemes, 3-*D* and CGT.

A. Distribution of Latent Sector Errors

LSEs refer to the incident sector errors that go undetected until the corresponding disk sectors are accessed [5]. Any data previously stored in the sector is lost. Therefore, it is necessary to consider the distribution of such incident LSEs since it is unpredictable and may affect the integrity verification of the whole hard disk. Here, a list of useful observations from [5] is shown. A total of 3.45% of 1.53 million hard disks developed LSEs over a period of 32 months. More than 80% of hard disks with LSEs have fewer than 50 errors. More importantly, there is significant locality in the occurrence of LSEs across logical sector addresses, and the hard disks exhibit high temporal locality of LSEs, says in "bursty" pattern. Schroeder et al. [12] further found that between 20% and 50% of all LSEs are located in the first 10% of the hard disk's logical sector space and between 20 to 60% of all errors have a neighbor within a distance of less than 10 sectors in logical sector space.

--*Observation 1*. Most of the numbers of LSEs for the hard disks are in dozens, whose allocations tend to be near each other or even consecutive, and tend to locate in the first part of the hard disks' logical sector space.

B. Distribution of Word Documents

Since Word document is a quite common digital format that is used by ordinary users frequently, it is a typical and important form of LPP data stored on a hard disk. As a preliminary work, we explore the allocation of Word documents to simulate the features of sector distribution of LPP documents.

Before looking into the details of sector distribution, we give an overview on basic structure of a hard disk [13]. Each hard disk is equipped with several flat disks called *platters*, each of which has two sides, the top and the bottom, accessed to by the two *heads*, respectively. Each platter is broken into concentric circles, *tracks*, each of which is further broken into *sectors*, which is the smallest physical storage unit (512 bytes per sector). In addition to the physical structure, the hard disks also provide the logical structure, which is generally named as *file system*. Different systems use different file allocation schemes to organize and control access to data on the hard disk. To balance the needs of efficient disk use and performance, *clusters*, or allocation units, are used to manage sectors (8 sectors per cluster in NTFS file system).

Our experiments are based on tens of the hard disks using NTFS file system, whose storage sizes are all 250GB. We search for all Word documents with "doc" and "docx" as file extension from normal users' hard disks, and find that on average there are about 3000 Word documents on a hard disk with an average size of 200K Bytes (i.e. 400 sectors or 50 clusters). Around 0.4% of all Word documents are further divided into several or more fragments, while all others are allocated in consecutive sectors/clusters. Note that our experiment does not consider those logically deleted Word documents which may still be left on the hard disk. According to the sector distribution of potential LPP word files, we have the following observation.

--*Observation 2*. The deletion of LPP data will affect consecutive sectors in the hard disk. The average number of consecutive sectors for Word documents of a normal user is 400.

C. Comparison of 3-D and CGT with Sectors Erased

In this subsection, we evaluate the 3-*D* hashing scheme and the CGT-based hashing scheme in the context of forensic investigation with possible deletion of LPP files in the presence of incident error sectors. We mainly investigate the number of affected sectors when there are sectors changed due to the random LSEs and the deletion of LPP documents.

In [14], the reliability of CGT-based hashing scheme with Shifted Transversal Design [15] (STD) has been analyzed with the number of errors varying from 1 to 10. The previous result shows that for a block containing 10^6 sectors, when the number of the error sectors is increased from 1 to 10, about 95% of total sectors can still be verified. Here, we increase the number of error sectors up to 50, and find that the CGT-based hashing scheme is almost useless since all the outcomes of group tests are corrupted due to the changed sectors. When a Word document containing 400 consecutive sectors is deleted to simulate the deletion of a LPP document, the CGT-based hashing scheme again cannot identify any normal sector.

As for 3-*D* scheme, we use Monte-Carlo method [16] to simulate the incident sector errors to investigate the number of affected sectors caused by LSEs, and also erase multiple Word documents to investigate the performance of the scheme. Note that a hard disk of 250GB with 488,392,065 sectors is taken as the device under test here. As shown in Figure 1, for the case of LSEs, when the number of the error sectors increases from 1200 to 18000, the average number of the affected sectors varies from 3 to 11480 and the curve is getting sharper. It means that the more the number of sectors is changed, the more the number of sectors in the cube is affected. And the number of

affected sectors increases very fast. However, when it turns to the case of the deletion of multiple Word documents (to simulate the deletion of LPP files), the number of affected sectors does not increase as much when the number of deleted files increases. Even when the number of deleted sectors for LPP files is increased to 18000, the number of the affected sectors is only 2362 which is far less than that of the case of random LSEs.

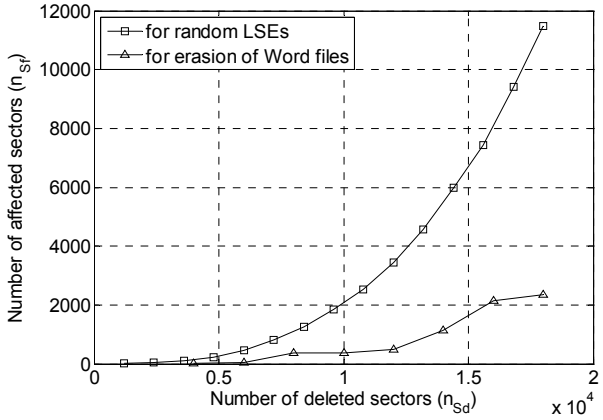

Figure 1. The number of affected sectors varies with the number of deleted sectors

--*Observation 3*. When the number of LSEs or the sectors of erased LPP documents increase, the performance of the CGT-based hashing scheme deteriorates greatly.

--*Observation 4*. When the number of LSEs increases, the performance of the 3-*D* hashing scheme becomes worse and worse. However, the decrease in performance is not as serious for the case of deletion of LPP files.

As a summary of the above observations, CGT is not appropriate when there are a large number of sectors erased from the hard disk, whatever they are consecutive or not. For 3-*D* scheme, it is not as bad as CGT and is a good candidate for further improvement. In the next section, we propose a "Dual Cube" hashing scheme based on the 3-*D* scheme to reduce the number of affected sectors.

IV. Dual Cube Hashing Scheme

The section first reviews the modified 3-*D* hashing scheme given in [11], then proposes the "Dual Cube" hashing scheme which can greatly enhance the performance of the integrity verification scheme by reducing the number of possibly affected sectors.

A. Review of the modified 3-D hashing scheme

In [6], the 3-*D* hashing scheme orders all the sectors on a hard disk in a 3-dimension pattern. Each sector s_m ($0 \le m \le N$-1) can be uniquely represented by three coordinates, denoted as $s_{x,y,z}$ where $1 \le x, y, z \le N^{1/3}$, and N is the total number of sectors of the hard disk.

Instead of computing one hash value for each sector, for each fixed (y, z) pair, it forms a hashing chain in X dimension using the sectors $s_{x,y,z}$ for all x's from 1 to $N^{1/3}$; for each fixed (x, z) pair, it forms a hashing chain in Y dimension using the sectors $s_{x,y,z}$ for all y's from 1 to $N^{1/3}$; for each fixed (x, y) pair, it forms a hashing chain in Z dimension using the sectors $s_{x,y,z}$ for all z's from 1 to $N^{1/3}$. The size of each chain is $N^{1/3}$ and one hash value is computed for each chain, as follows (see Figure 2).

$$VX_{y,z} = Hash(s_{1,y,z} \| s_{2,y,z} \| \cdots \| s_{N^{1/3},y,z}),$$

$$VY_{x,z} = Hash(s_{x,1,z} \| s_{x,2,z} \| \cdots \| s_{x,N^{1/3},z}),$$

$$VZ_{x,y} = Hash(s_{x,y,1} \| s_{x,y,2} \| \cdots \| s_{x,y,N^{1/3}}).$$

Then, all the hash values will be stored in a secure place for later comparison. After some time when the hard disk integrity checking is required, all the hash values will be recalculated using the same 3-*D* scheme on the hard disk sectors, then compared to the originally stored ones.

However, the problem of this scheme is that it will fail to verify the integrity of a normal sector when there is at least one bad sector on each chain to which the normal sector belongs, i.e., the normal sector happens to be arranged at the intersection of three "corrupted" chains in a cube such that it is tainted. For example, in Figure 2, the scheme fails to verify the integrity of the sector $s_{x,y,z}$ (a black point) for there is a bad sector $s_{i0,j0,k0}$ (a crossed point) with ($j_0 = y$; $k_0 = z$) in X dimension, a bad sector $s_{i1,j1,k1}$ (a crossed point) with ($i_1 = x$; $k_1 = z$) in Y dimension, and a bad sector $s_{i2,j2,k2}$ (a crossed point) with ($j_2 = y$; $k_2 = z$) in Z dimension simultaneously. In the case of LPP documents deletion, this problem becomes more serious because a lot of sectors are artificially turned into "bad" resulting in more affected sectors that will fail the verification.

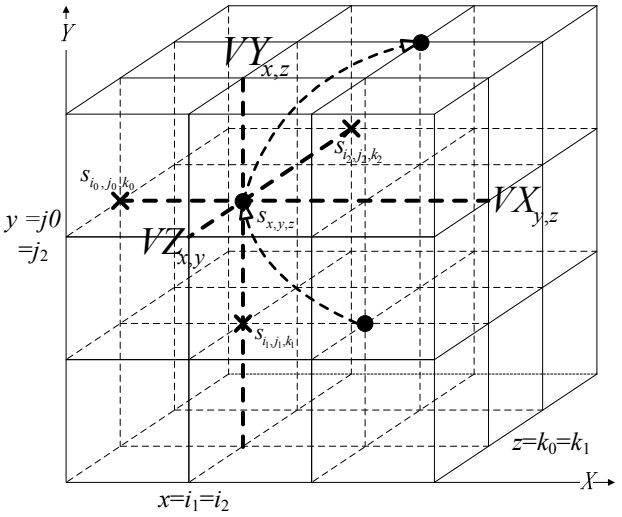

Figure 2. A sample structure of 3-*D* hashing scheme

In [11], a modified 3-*D* scheme is proposed to accommodate this scheme to the LPP application. The idea is to locate the affected sectors, S_f, according to the list of sectors to be deleted, S_d, then form these affected sectors as an additional hash chain (the dashed line with arrow) such that these sectors can be verified separately later. And the authors expect that the number of deleted sectors may not be a very big number, as well as the number of affected sectors. However, as pointed out in the Introduction section, the assumption of knowing which sectors to be deleted is not realistic or both the suspect/his legal

representative and the forensic investigator have to sit down for long hours to go through each file to decide if it is LPP related.

B. Our proposed "Dual Cube" hashing scheme

Although the modified 3-*D* hashing scheme provides a good solution to the hard disk integrity checking with LPP documents deletion, the performance may deteriorate when the number of LPP documents increases. With the increase in the number of sectors to be deleted, the number of affected sectors also increases rapidly. For example, in our experiments, when the number of deleted sectors for LPP documents is increased to 120000, the average number of affected sectors becomes 551036, and exceeds the number of deleted sectors by about 5 times. In this case, how to check the integrity of the additional chain effectively and efficiently becomes another problem in addition to other possible problems such as latent sector error, hash corruption caused by one bit loss, the cost of computation and storage, and etc. [7].

Therefore, in our proposed scheme, we try to reduce the number of affected sectors to a minimal level by using a "Dual Cube" hashing method, such that to avoid the lengthy additional chain, even when the number of deleted sectors for LPP documents is huge. The idea of "Dual Cube" hashing method is as follows.

HASH VALUES GENERATION: When the investigator wants to generate integrity information of the hard disk, the procedure of "Dual Cube" hashing scheme is as follows:

(1) First, as the original 3-*D* scheme does, the sectors in the hard disk will be arranged into a cube according to the normal sequence of sectors, which is denoted as SEQ^{1st} *= (0, 1, 2, ..., N-1)*.

(2) With this cube, a hash value for every chain in every direction for every sector will be computed. Note that there will be 3 hash values correlated to every sector, denoted as $VX^{1st}_{y,z}$, $VY^{1st}_{x,z}$ and $VZ^{1st}_{x,y}$. After computation, all the hash values will be stored securely for further purpose.

(3) After step (2), the sequence of the sectors will be shuffled and a new sequence, denoted as SEQ^{2nd} *= (N/2, N/2+1, N/2+2, ..., N, 0, 1, 2, ..., N/2-1)* will be generated. Using SEQ^{2nd}, the sectors will be rearranged to form a new cube. This operation is equivalent to rotating the cube as rotating a Rubik's cube.

(4) Since now a certain sector in the second cube will have three new chains which are different with those in the first cube, we need to compute a new set of hash values for the new chains for every sector. And the new set of hash values for the sector can be denoted as $VX^{2nd}_{y,z}$, $VY^{2nd}_{x,z}$ and $VZ^{2nd}_{x,y}$. These hash values will also be stored for integrity verification.

HASH VALUES VERIFICATION: After deleting LPP documents, to verify the integrity of the remaining sectors, the investigator can works as follows:

(1) With SEQ^{1st} *= (0, 1, 2, ..., N-1)*, the investigator can arrange the sectors into a cube same as the first cube in hash values generation stage and compute a set of hash values using this cube: $VX^{1st}_{y,z}{}'$, $VY^{1st}_{x,z}{}'$ and $VZ^{1st}_{x,y}{}'$.

(2) Comparing the hash values set of $VX^{1st}_{y,z}{}'$, $VY^{1st}_{x,z}{}'$ and $VZ^{1st}_{x,y}{}'$ with the stored ones, $VX^{1st}_{y,z}$, $VY^{1st}_{x,z}$ and $VZ^{1st}_{x,y}$, the investigator will identify the sectors as unmodified or not and get a collection of sectors which fail in the verification. The collection can be denoted as $S^{1st}_{nv} = \{S_d, S^{1st}_f\}$, where S_d denotes the sectors deleted for LPP documents, and S^{1st}_f denotes the sectors affected by the deleted sectors in the first cube.

(3) Using SEQ^{2nd}, the sectors will be rearranged to form a new cube as the second cube in hash values generation stage, and the corresponding hash values set of $VX^{2nd}_{y,z}{}'$, $VY^{2nd}_{x,z}{}'$ and $VZ^{2nd}_{x,y}{}'$ will be computed.

(4) As step (2) does, the investigator compares the hash values set of $VX^{2nd}_{y,z}{}'$, $VY^{2nd}_{x,z}{}'$ and $VZ^{2nd}_{x,y}{}'$ with the stored ones, $VX^{2nd}_{y,z}$, $VY^{2nd}_{x,z}$ and $VZ^{2nd}_{x,y}$, and results in a new collection of sectors which fail in the verification for the second cube. The collection can be denoted as $S^{2nd}_{nv} = \{S_d, S^{2nd}_f\}$, where S^{2nd}_f represents the sectors affected by the deleted sectors in the second cube.

(5) Since both collections, $S^{1st}_{nv} = \{S_d, S^{1st}_f\}$ and $S^{2nd}_{nv} = \{S_d, S^{2nd}_f\}$, include S_d, through comparing these two collections, the investigator can locate the intersection point of these collections and the intersection will be almost the same as S_d except the number of the items in S_d is extremely huge.

The experimental results for "Dual Cube" hashing method versus the original 3-*D* scheme are shown in Figure 3. The capacity of the hard disk under test is 250GB with 488,392,065 sectors. 3000 Word documents are prepared for this experiment and each document occupies 400 continuous sectors in the hard disk on average. When the number of deleted sectors for LPP documents is increased from 8000 to 120000, the number of affected sectors for 3-*D* scheme is increased from 100 to 551036 on average, i.e., the number of affected sectors is increased 5510 times. At the mean time, the number of affected sectors for "Dual Cube" is only about 1000 even there are 120000 sectors are deleted. This result illustrates that the "Dual Cube" method is more reliable in integrity checking at sector level than the previous 3-*D* scheme from the viewpoint of practice.

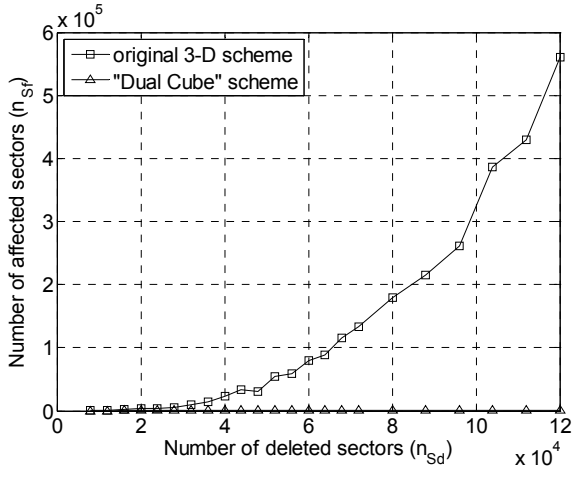

Figure 3. The number of affected sectors using "Dual Cube" method and that using original 3-*D* scheme versus the increment of the number of deleted sectors.

C. *Mathematical analysis*

In this section, we will try to deduce the probability of a normal sector which becomes unverifiable as it is affected by n_{Sd} randomly deleted sectors in the "Dual Cube" scheme, such that we can estimate the number of affected sectors.

(1) At the first step, we try to deduce the probability of a normal sector affected by n_{Sd} randomly deleted sectors in the first cube. Assume that there is a hard disk drive containing *N* sectors and we distribute the sectors into a cube. For this cube, the length of each dimension should be $R=N^{1/3}$.

Instead of calculating the probability of affected sector directly, we first calculate the probability for each chain in *d1* dimension which will be affected by the deleted sectors. The idea is as follows. If there is only one sector deleted, the probability for each chain becomes unverifiable will be $1/N^{2/3}$ because there are totally ($N^{1/3}*N^{1/3}$) chains in *d1* dimension, and the probability for each chain remains unaffected will be ($1-1/N^{2/3}$). Therefore, if there are n_{Sd} independently deleted sectors, the probability for each chain remains unaffected will be $(1-1/N^{2/3})^{n_{Sd}}$, and the probability for each chain becomes unverifiable will be $1-(1-1/N^{2/3})^{n_{Sd}}$. Since each sector in the cube will uniquely belong to a certain chain in *d*1 dimension, the probability of a normal sector which will become unverifiable in *d1* dimension is equivalent to the probability for each chain becomes unverifiable due to the deleted sectors, denoted as $p_{d1}=1-(1-1/N^{2/3})^{n_{Sd}}$. Similarly, the probability for the other two dimensions can be denoted as $p_{d2}=1-(1-1/N^{2/3})^{n_{Sd}}$ and $p_{d3}=1-(1-1/N^{2/3})^{n_{Sd}}$. A normal sector in the cube will be affected only when all of those three chains become unverifiable due to the deleted sectors. Thus, the probability of a sector becomes unverifiable will be

$$p_{af} = p_{d1} * p_{d2} * p_{d3} * (1-n_{Sd}/N),$$

where the term $(1-n_{Sd}/N)$ means that the deleted sectors should not be count into the group of affected sectors.

(2) Then we can consider the probability in "Dual Cube" scheme. As the second cube will be different with the first one, all the three chains correlated to a certain sector will be changed, as well as the constraint relationship. Therefore, we get three new probabilities:

$$p_{d1}^{2nd} = 1-(1-1/N^{2/3})^{n_{Sd}},$$
$$p_{d2}^{2nd} = 1-(1-1/N^{2/3})^{n_{Sd}} \text{ and}$$
$$p_{d3}^{2nd} = 1-(1-1/N^{2/3})^{n_{Sd}}$$

And the probability of a sector becomes unverifiable in "Dual Cube" scheme will be

$$p_{af}^{Dual\ Cube} = p_{d1} * p_{d2} * p_{d3} * p_{d1}^{2nd} * p_{d2}^{2nd} * p_{d3}^{2nd} * (1-n_{Sd}/N)$$

(3) These formula of probabilities can be used to estimate the number of affected sectors in 3-*D* scheme by:

$$n_{af}^{\exp} = (N-n_{Sd}) * p_{d1} * p_{d2} * p_{d3},$$

where $n_{af}^{\exp}$ means the expectation of the number of affected sectors. And the number of affected sectors in "Dual Cube" scheme will be:

$$n_{af}^{\exp} = (N-n_{Sd}) * p_{d1} * p_{d2} * p_{d3} * p_{d1}^{2nd} * p_{d2}^{2nd} * p_{d3}^{2nd} *$$

Figure 4 shows the estimated curve from the formulas and the estimated curve fits well with the experimental results.

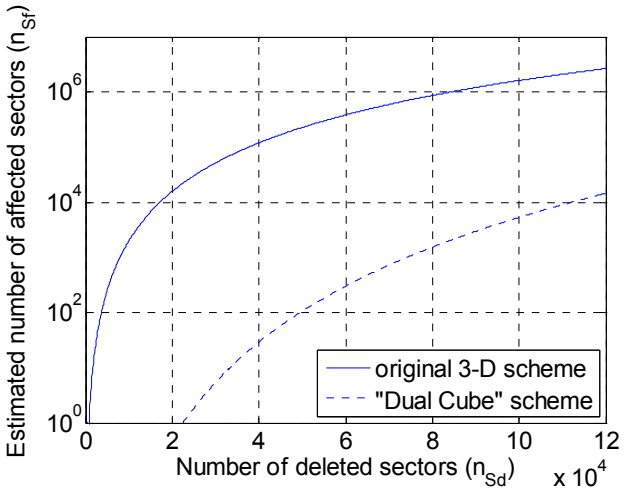

Figure 4. The estimated number of affected sectors using "Dual Cube" method and that using original 3-*D* scheme versus the increment of the number of deleted sectors.

D. *Enhance accuracy and reliability using multiple cubes*

Although we expect that the number of LPP documents, i.e. the number of deleted sectors for LPP documents, may not be a very big number, it is still possible, especially when the capacity of a hard disk is huge and is getting more and more huge. If the number of deleted sectors is kept increasing, using the "Dual Cube" method will also meet its limitation. In the extreme case, if there are too many sectors deleted and the number of affected sectors is still large even after the "Dual Cube" hashing method is deployed, multiple cubes can be used to keep successfully verifying these affected sectors from the practical viewpoint.

For example, in the above experiments, assume that there are 600,000 sectors deleted, using the original 3-*D* hashing scheme will cause the failure of verifying 28,294,060 sectors,

and the number of affected sectors using "Dual Cube" hashing scheme is still 1,675,031. In this case, deploying a third cube as a complement is a feasible solution to further reduce the number. In our experiment, the number of affected sectors under this test is only 103,147. Multiple cubes may be used to handle the case if more sectors are expected to be deleted. Of course, there will be a trade-off on the number of hash values to be stored.

V. Conclusions

This paper evaluates the existing hashing methods proposed for solving the LPP integrity problem based on two types of changes in the sectors: incident error sectors and deletion of LPP files. The evaluation results found that the 3-*D* scheme is acceptable while the CGT-based scheme is not good enough. Then, we proposed the "Dual Cube" hashing scheme which can achieve better performance. The scheme was also analyzed theoretically for random sector errors. As one of the future research directions to the problem, we would conduct a theoretical analysis on this scheme for LPP sector deletion. More comprehensive experiments should be carried to understand the sector distribution of LPP data in order to design a better scheme. A better scheme which uses fewer hash values with fewer affected sectors is always desirable. The proposed methodology is being used by a law enforcement department in some real cases. The result will be reported in the full paper.

References

[1] F. Yu, Y. Lei, Y. Wang, and Z. Lu, "Robust Image Hashing Based on Statistical Invariance of DCT Coefficients," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 1, pp. 286-291, 2010.

[2] Association of Chief Police Officers (ACPO) - Good practice guide for computer based electronic evidence, (http://www.dataclinic.co.uk/ACPO Guide v3.0.pdf); accessed on 31st January, 2008.

[3] Anti Cartel Enforcement Manual, International Competition Network, April 2006, http://www.internationalcompetitionnetwork.org/media/library/conference_5th_capetown_2006/DigitalEvidenceGathering.pdf; accessed on 30th January, 2008.

[4] K. P. Chow, C. F. Chong, K. Y. Lai, L. C. K. Hui, K. H. Pun, W. W. Tsang, and H. W. Chan, "Digital evidence search kit," in *Proc. of the First International Workshop on Systematic Approaches to Digital Forensic Engineering (SADFE'05)*, 2005, pp. 187-194.

[5] L. N. Bairavasundaram, G. R. Goodson, S. Pasupathy, and J. Schindler, "An Analysis of Latent Sector Errors in Disk Drives," in *SIGMETRICS'07*, San Diego, California, USA, June 12–16, 2007, pp. 289-300.

[6] Z. L. Jiang, L. C. K. Hui, and S. M. Yiu, "Analysis of K-Dimension Hashing Scheme to Improve Disk Sector Integrity," in *Proc. of the 4th Annual IFIP WG 11.9 International Conference on Digital Forensics (ICDF 2008)*, 2008, pp. 33-44.

[7] J. Fang, Z. L. Jiang, S. M. Yiu, and L. C. K. Hui, "Hard Disk Integrity Check by Hashing with Combinatorial Group Testing," in *Proc. of the 2nd International Computer Science and its applications*, 2009 (CSA 09), 10-12 Dec., 2009, pp. 1-6.

[8] L. Chen and G. Y. Wang, "An Integrity Check Method for Fine-Grained Data," *Journal of Software*, 20(4), 2009, pp. 902-909.

[9] L. Chen, X. L. Fang and G. Y. Wang, "One Error Integrity Indication Codes and Performance Analysis," *Computer Science*, 36(6), 2009, pp. 97-100.

[10] P. E. Nygh, P. Butt, *Butterworths Concise Australian Legal Dictionary*, Sydney: Butterworths, 1997.

[11] F. Y. W. Law, P. K. Y. Lai, Z. L. Jiang, R. S. C. Ieong, M. Y. K. Kwan, K. P. Chow, L. C. K. Hui, and S. M. Yiu, "Protecting digital legal professional privilege (LPP) data," in *Proc. of the 3rd International Workshop on Systematic Approaches to Digital Forensic Engineering (IEEE/SADFE-2008)*, 2008, pp. 91-101.

[12] B. Schroeder, S. Damouras and P. Gill, "Understanding Latent Sector Errors and How to Protect Against Them," *ACM Transactions on Storage*, Vol. 6, No. 3, September 2010, pp. 9:1-9:23.

[13] C. M. Kozierok, "The PC Guide," 2001. Available: *http://www.pcguide.com*.

[14] J. Fang, Z. L. Jiang, S.M. Yiu, Lucas C.K. Hui, "Checking key integrity efficiently for high-speed quantum key distribution using combinatorial group testing," *Optics Communications*, 284(3), 2011, pp. 531-535. doi:10.1016/j.optcom.2010.08.066.

[15] N. Thierry-Mieg, "A new pooling strategy for high throughput screening: the shifted transversal design," *Bmc Bioinformatics*, 7, 2006, pp. 13.

[16] N. Metropolis, S. Ulam, "The Monte Carlo Method," *Journal of the American Statistical Association,* 44(247), 1949, pp. 335-341. doi:10.2307/2280232